Practical6: Useful software utilities for computational genomics

Shamith Samarajiwa

September 2017

Contents

0.0.1 Setup

bash

```
pwd

cd "~/Course_Materials/ChIPSeq/Materials/Practicals"
mkdir utils
cd utils
```

R

```
# show current directory
getwd()

#set new working directory in R
setwd("/home/participant/Course_Materials/ChIPSeq/Materials/Practicals/utils")

# shows directory contents
dir()
```

Install and load packages needed for the tutorial. Uncomment install commands if your computer doesn't have the packages.

R

```

#load libraries

library("GenomicRanges")
library("TxDb.Hsapiens.UCSC.hg38.knownGene")
library("EnsDb.Hsapiens.v86")
library("org.Hs.eg.db")
library("ChIPseeker")
library("ChIPpeakAnno")
library ("rtracklayer")
```

Read the peak files into GRanges objects.

A complete collection of peak files are in the Macs2 folder. We will read the (Excel) xls file for one of the TP73 replicate into a R dataframe.

R

```
library("GenomicRanges")

peakfile1 <- "~/Course_Materials/ChIPSeq/Preprocessed/Peaks/TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks
peakfile2 <- "~/Course_Materials/ChIPSeq/Preprocessed/Peaks/TAp73alpha_r1.fastq_trimmed.fastq_sorted_peaks


#generic code ..adapt this to process the two peakfiles or look at peak calling practical from day3
# or readPeakFile from ChIPseeker

peaks_DF <- read.delim2(peakfile, comment.char="#")
peaks_DF[1:3,]



library(GenomicRanges)
peaks_GR <- GRanges(
  seqnames=peaks_DF1[,"chr"],
  IRanges(peaks_DF1[,"start"],
  peaks_DF1[,"end"]
  )
)


df <- data.frame(seqnames=seqnames(peaks_GR),
  starts=start(peaks_GR)-1,
  ends=end(peaks_GR),
  names=c(rep(".", length(peaks_GR))),
  scores=c(rep(".", length(peaks_GR))),
  strands=strand(peaks_GR))

# change the name according to sample

write.table(df, file="TFname_ReplicateName.bed", quote=F, sep="\t", row.names=F, col.names=F)
```

1 Bedtools

These commands need to be run in a bash shell. Create any bed files needed using the previous code chunk.

cd into working directory in bash shell

```
pwd
cd /home/participant/Course_Materials/ChIPSeq/Materials/Practicals/utils
```

Bedtools is a command-line tool. To bring up the help, just type

```
bedtools
```

Use bedtools with the appropriate subcommand.

Ex:

```
bedtools intersect
bedtools merge
bedtools subtract
```

Version

```
bedtools --version
```

- Main functionality

1.1 Intersections

- Default behaviour of intersect is to reports the intervals that represent overlaps between two files. To demonstrate, let's identify all of the chip peaks that overlap with CpG islands.

```
# Download the CpG island track from UCSC table browser and name it hg38_CpG_Islands.bed

bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed | \
head -5

#write to file

bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed > \
TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks_CPG.bed
```

- Reporting the original feature in each file
- The -wa (write A) and -wb (write B) options allow one to see the original records from the A and B files that overlapped. As such, instead of not only showing you where the intersections occurred, it shows you what intersected.

```
bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed -wa -wb |
head -5
bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed -wa -wb >
TAp73alpha_r2_CPG_all.bed
```

- Count of overlaped nucleotides.

```
bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed -wo | hea
bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed -wo > \
TAp73alpha_r2_CPG_overlap_nt.bed
```

- Count of overlapping features.

```
bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed -c | head
bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed -c > \
TAp73alpha_r2_CPG_overlap_ft.bed
```

- Find features that DO NOT overlap

bash

```
bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed -v | head
bedtools intersect -a TAp73alpha_r2.fastq_trimmed.fastq_sorted_peaks.bed -b hg38_CpG_Islands.bed -v > \
TAp73alpha_r2_CPG_overlap_notoverlap.bed
```

- Find the complement

bash

```
pwd

bedtools sort hg38_CpG_Islands.bed

bedtools complement -i hg38_CpG_Islands.bed \
-g ~/Course_Materials/Introduction/SS_DB/Reference/STAR/hg38_chr3.genome > non_cpg_regions.bed
```

Sort and Merge

```
# sort

# sort -k1,1 -k2,2n foo.bed > foo.sort.bed
```

2 ChIPseeker

- Generate tag matrix around putative promoter region

```

library(TxDb.Hsapiens.UCSC.hg38.knownGene)
library(ChIPseeker)
library(clusterProfiler)


txdb <- TxDb.Hsapiens.UCSC.hg38.knownGene
promoter <- getPromoters(TxDb=txdb, upstream=3000, downstream=3000)

# use GRanges object generated in the previous section

tagMatrix <- getTagMatrix(peaks_GR, windows=promoter)
```

Heatmap of ChIP binding to TSS regions

```
tagHeatmap(tagMatrix, xlim=c(-3000, 3000), color="red")
```

plotAnnoBar(peakAnno)

Average Profile of ChIP peaks binding to TSS region

```
plotAvgProf(tagMatrix, xlim=c(-3000, 3000), xlab="Genomic Region (5'->3')", ylab = "Read Count Frequency")

# or use this single function (without generating a tag matrix)

plotAvgProf2(peaks_GR, TxDb=txdb, upstream=3000, downstream=3000,
             xlab="Genomic Region (5'->3')", ylab = "Read Count Frequency")

# confidance intervals by resampling
plotAvgProf(tagMatrix, xlim=c(-3000, 3000), conf = 0.95, resample = 1000)
```

Peak annotation by genomic features

```
peakAnno <- annotatePeak(peaks_GR, tssRegion=c(-3000, 3000), TxDb=txdb, annoDb="org.Hs.eg.db")
```

Plot genomic feature profiles of peaks

```
#pie chart
plotAnnoPie(peakAnno)

#bar plot
plotAnnoBar(peakAnno)
```

Venn plot

```
upsetplot(peakAnno, vennpie=TRUE)
```

3 Visualize distribution of TF-binding loci relative to TSS

```
plotDistToTSS(peakAnno, title="Distribution of transcription factor-binding loci\nrelative to TSS")
```

4 Annotating genes using ChIPpeakAnno

```
library("EnsDb.Hsapiens.v86")
library("ChIPpeakAnno")

# Prepare annotation data with toGRanges

annoData <- toGRanges(EnsDb.Hsapiens.v86)
annoData[1:2]

# Annotate the peaks with annotatePeakInBatch

## keep the seqnames in the same style
seqlevelsStyle(peaks_GR) <- seqlevelsStyle(annoData)

## do annotation by nearest TSS
anno <- annotatePeakInBatch(peaks_GR, AnnotationData=annoData)
anno[1:2]
```

5 A pie chart can be used to demonstrate the overlap features of the peaks.

```
pie1(table(anno$insideFeature))
```

6 Additional annotation

```
library(org.Hs.eg.db)
anno <- addGeneIDs(anno, orgAnn="org.Hs.eg.db",
                   feature_id_type="ensembl_gene_id",
                   IDs2Add=c("symbol"))
head(anno)
```

```
df <- data.frame(seqnames=seqnames(anno),
  starts=start(anno)-1,
  ends=end(anno),
  names=c(rep(".", length(anno))),
  scores=c(rep(".", length(anno))),
  strands=strand(anno))

write.table(df, file="TP73_anno.bed", quote=F, sep="\t", row.names=F, col.names=F)
str("TP73_anno.bed")
```

```
sessionInfo()
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] BiocStyle_2.4.1
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.4.1  backports_1.1.0 magrittr_1.5    rprojroot_1.2
##  [5] tools_3.4.1     htmltools_0.3.6 yaml_2.1.14     Rcpp_0.12.12
##  [9] stringi_1.1.5   rmarkdown_1.6   knitr_1.17      stringr_1.2.0
## [13] digest_0.6.12   evaluate_0.10.1
```